

CLAIMS

1. A method, comprising:

receiving, from a client, a request for a document;

assigning a session ID to uniquely identify a session established with the client;

5 generating a time stamp;

associating the time stamp with the session ID;

serving the document along with a time stamp and an executable script to the client, the executable script being configured to return the time stamp when the document is rendered on the client;

10 receiving the time stamp from the client;

deriving a document render time from the time stamp, the document render time being indicative of a time period from when the request for the document is generated at the client to when the document is rendered at the client;

15 logging the document render time in association with the session ID that is associated with the time stamp from which the document render time is associated; and

determining an average render time per user as a function of one or more logged document render times for a common session ID.

2. A method as recited in claim 1, wherein the document comprises a web

20 page.

3. A method as recited in claim 1, wherein the deriving comprises computing a time period from the time stamp and a current time, the time period representing the document render time.

4. A method as recited in claim 1, wherein multiple document render times for a given session ID are logged, and the determining further comprises computing an average render time per user by averaging the document render times for the common session ID.

5

5. A method as recited in claim 1, further comprising sending the session ID together with the time stamp to the client and receiving the session ID and the time stamp back from the client upon execution of the script.

10 6. A method as recited in claim 1, further comprising logging the average render time.

7. A method, comprising:

receiving multiple requests for information from a client during a session;

15 sending responses to the client, each response containing a value and an executable program, the executable program being configured to return the value when the information is presented on the client;

receiving the values from the client in response to execution of the executable program contained with each said response; and

20 ascertaining a metric representative of an average time to display requested information as a function of the values received for the session.

8. A method as recited in claim 7, wherein the value is derived from a monotonically increasing source and the ascertaining comprises:

25 comparing the value with a current value from the monotonically increasing source;

deriving a time to display result as a function of the value and the current value;

collecting multiple time to display results; and
deriving the average time to display as a function of the time to display results.

9. A method as recited in claim 7, wherein the value is a time stamp and the
5 ascertaining comprises:

comparing the time stamp with a current time;
deriving a time to display result as a difference between the time stamp and the
current time;
collecting multiple time to display results; and
10 deriving the average time to display as a function of the time to display results.

10. A method as recited in claim 7, further comprising logging the average
time to display.

15 11. A method, comprising:

receiving multiple requests for information from multiple clients during various
user sessions;
sending responses to the clients, each response containing a value and an
executable program, the executable program being configured to return the value when
20 the information is presented on the client;
receiving the values from the clients in response to execution of the executable
program contained with each said response; and
ascertaining an average render time per unique user session as a function of the
values received for the clients.

12. A server system, comprising:

a server to receive one or more requests for renderable documents from a client during a session, the server being configured to serve the documents in response to the requests, each served document being accompanied by a script and a time value;

5 individual scripts being configured to execute in response to their corresponding documents being rendered on the client such that when executed, the individual scripts return the time value to the server; and

a time-to-render monitor to determine an average render time based on returned time values for the session.

10

13. A server system as recited in claim 12, wherein the document comprises a web page.

14. A server system as recited in claim 12, wherein the time-to-render monitor
15 approximates a render time for each returned time value by calculating a difference between the time stamp and a current time.

15. A server system as recited in claim 14, further comprising a log to store the approximated render times, the average render time being derived from an average of the
20 approximated render times for a common session ID.

16. A server system as recited in claim 12, wherein each served document is further accompanied by the session ID and the script, when executed, returns the session ID along with the time value.

25

17. An architecture, comprising:

a server module resident at a first computer to serve one or more documents from the first computer to a second computer during a session, each document being accompanied by a time stamp and an executable script that executes when the document is rendered at the second computer to return the time stamp to the first computer; and

5 a render time measurement module resident at the first computer to receive the time stamps returned from the second computer and to derive an average document render time from the time stamps received for the session.

10 18. An architecture as recited in claim 17, wherein the document comprises a document written in a markup language.

19. An architecture as recited in claim 17, wherein the render time measurement module approximates the document render time by calculating a difference 15 between the time stamp and a current time.

20. An architecture as recited in claim 17, wherein the render time measurement module derives a document render time from each time stamp, the document render time being representative of a time period from when a request for the 20 document is first generated at the second computer to when the document is rendered at the second computer, the .

21. An architecture as recited in claim 17, further comprising a browser resident at the second computer to render the document.

25

22. A server computer comprising the architecture as recited in claim 17.

23. Computer-readable media having computer-executable instructions that, when executed by one or more processing units, direct one or more computing devices to:

5 serve a renderable document together with a time stamp and an executable program from a server to a client, the time stamp being associated with a session ID identifying a session with the client;

execute, upon rendering of the document at the client, the executable program at the client to return the time stamp to the server; and

10 derive a document render time from the time stamp, the document render time being indicative of a time period from when the request for the document is generated at the client to when the document is rendered at the client; and

determine an average document render time per user as a function of multiple document render times derived over time for a common session ID.

15 24. Computer-readable media as recited in claim 23, further comprising computer-executable instructions that, when executed by one or more processing units, direct one or more computing devices to compute the document render time from a function of the time stamp and a current time.

20 25. Computer-readable media as recited in claim 23, further comprising computer-executable instructions that, when executed by one or more processing units, direct one or more computing devices to serve the session ID along with the time stamp and to return the session ID with the time stamp back to the server.